Editors' Suggestion

Folding lattice proteins confined on minimal grids using a quantum-inspired encoding

Anders Irbäck,[1,*] Lucas Knuthson,[1] and Sandipan Mohanty[2]

[1] *Computational Science for Health and Environment (COSHE), Centre for Environmental and Climate Science, Lund University, 223 62 Lund, Sweden*

[2] *Jülich Supercomputing Centre, Forschungszentrum Jülich, D-52425 Jülich, Germany*

(Received 25 June 2025; accepted 1 September 2025; published 1 October 2025)

Steric clashes pose a challenge when exploring dense protein systems using conventional explicit-chain methods. A minimal example is a single lattice protein confined on a minimal grid, with no free sites. Finding its minimum energy is a hard optimization problem, with similarities to scheduling problems. It can be recast as a quadratic unconstrained binary optimization (QUBO) problem amenable to classical and quantum approaches. We show that this problem in its QUBO form can be swiftly and consistently solved for chain length 48, using either classical simulated annealing or hybrid quantum-classical annealing on a D-Wave system. In fact, the latter computations required about 10 s. We also test linear and quadratic programming methods, which work well for a lattice gas but struggle with chain constraints. All methods are benchmarked against exact results obtained from exhaustive structure enumeration, at a high computational cost.

DOI: 10.1103/8n7p-7lh2

I. INTRODUCTION

Recent years have seen big advances in predicting the native structure of folded proteins, through the development of evolutionary-informed machine learning techniques [1]. Still, other biomolecular problems, such as protein aggregation and phase separation, remain challenging. These problems often concern dense systems. Exploring dense biomolecular systems by conventional simulations is computationally demanding. A major obstacle is steric clashes, which obstruct chain motions. This problem is present even in simplified models where the molecules are represented as self-avoiding chains on a lattice.

A simple yet extreme example is that of a single lattice protein confined on a maximally compact grid, where no site is left unoccupied. Studies of maximally compact lattice models can provide insights into the sequence-structure relationship of proteins [2–6]. However, for maximally compact systems, standard Monte Carlo methods are impractical, due to difficulties in proposing new feasible states. To find the minimum-energy structure of a given sequence, most studies have instead relied on exhaustive enumerations of all states, which is an option for short chains.

Another approach to the challenge posed by steric clashes is to switch from the original explicit-chain representation to an expanded binary representation, and thus recast the energy minimization problem as a quadratic unconstrained binary optimization (QUBO) task. In the binary system, chain structure is enforced through soft constraints, implemented as energy penalties [7–9]. Mapping hard discrete optimization problems onto binary systems has a long history [10,11], and has seen a revival with the advent of quantum optimization [12–15] and various classical Ising machines [16]. The approach has been applied to lattice protein [7–9] and homopolymer [17,18] problems, as well as to somewhat related scheduling tasks [19–21] that involve restrictions similar to steric clashes.

In this paper, we examine the utility of a binary representation in the determination of minimum-energy structures of maximally compact lattice proteins. We explore and compare three methods for optimizing the binary system. The first two are classical simulated annealing (SA) [22] and hybrid quantum-classical annealing (HA) as offered by D-Wave [23], both of which solve a QUBO variant of the problem with soft constraints. In addition, we perform calculations with the Gurobi optimizer (GO) [24], which works with hard constraints and combines linear and quadratic programming with heuristics. The results obtained with these three methods are evaluated against exact data, which we generate using much more time-consuming exhaustive enumeration techniques.

We consider lattice proteins in three dimensions, with a 20-letter amino acid alphabet and pairwise interaction energies given by the statistical Miyazawa-Jernigan potential [25,26]. Specifically, we search for minimum-energy structures for six different amino acid sequences with length $N = 48$ (Table I) on a 4×4×3 lattice. This was the largest problem size for which exact results could be generated with the resources at our disposal.

The results presented below show that the two QUBO-based approaches, HA and SA, are able to quickly and reliably solve the energy minimization task for these sequences. With

[*] Contact author: anders.irback@cec.lu.se

TABLE I. Amino acid sequences studied, in one-letter code. All six sequences fold to one of two topologies called A and B with low and high complexity, respectively [27]. The minimum energy, $E_{\text{MJ}}^{\text{min}}$, is indicated.

No.	Topology	Sequence	$E_{\text{MJ}}^{\text{min}}$
1	A	`FRTRPLNHDF YNYKIWEPFK PADFPKAWDR MLDHVWDSMA SWGHQHCS`	−25.85
2	A	`CDLPPFTYRH HGNDFWKNYE MIKHWDLWRD MFRAFWSDPV KASPHQAS`	−25.92
3	A	`FRTPWVSHQF YAYKLMEHFK WGDFCRNMDK WIDSLPDRWN PAPHDHAS`	−26.09
4	B	`KDKIHFRMNY GYPAWDAQSV KDLTCPRDWH FPHMRDPSHN WELAFFWS`	−25.87
5	B	`ENDVTMDMDP SPCLFRIHNL PRAHSFDRFG WHQFDKYHYK WKWAWAPS`	−26.15
6	B	`EHDAQLDFDW SRWTWHGRNS YHAPAMYRWP VHDMDKPNPK FKIFFLCS`	−26.24

HA, it took the order of 10 s to find the solutions, whereas the exhaustive enumeration code took the order of 10 h to run.

II. METHODS

A. Biophysical model

We consider a coarse-grained protein model, where the protein is represented as a self-avoiding chain of N amino acids, or beads, on a lattice. The interaction potential is pairwise additive and stipulates that two amino acids interact only if they are nearest neighbors on the lattice but not along the chain. Let $\mathbf{r}_i$ and a_i denote, respectively, the position and type of amino acid i ($i = 1, \dots, N$). The interaction energy, E_{MJ}, can then be written as

$$E_{\text{MJ}} = -\sum_{1\leqslant i<j-1\leqslant N} C(a_i, a_j)\Delta(\mathbf{r}_i, \mathbf{r}_j), \tag{1}$$

where $\Delta(\mathbf{r}_i, \mathbf{r}_j) = 1$ if $\mathbf{r}_i$ and $\mathbf{r}_j$ are nearest neighbors on the lattice, and $\Delta(\mathbf{r}_i, \mathbf{r}_j) = 0$ otherwise.

In prior work, we explored the application of HA to fold [9] and design [28] lattice proteins in a minimal model of this form, namely, the two-letter HP model on a two-dimensional (2D) square lattice [2]. In that model, the amino acids are either hydrophobic (H) or polar (P). The contact energy is given by $C(a_i, a_j) = 1$ if $a_i = a_j =$ H, and $C(a_i, a_j) = 0$ otherwise.

In the present paper, we use a 20-letter amino acid alphabet, a three-dimensional cubic lattice, and contact energies given by the statistically derived Miyazawa-Jernigan interaction matrix (Table VI of Ref. [25]). Specifically, we minimize E_{MJ} [Eq. (1)] for six different 48–amino acid sequences (Table I) confined on a minimal 4×4×3 grid, without any free sites. Despite the restriction to maximally compact states, there are $\approx 1.3\times10^{11}$ possible conformations available for these chains [29,30]. These same six sequences have been studied previously by Faísca and Plaxco, using an unrestricted grid [27]. The sequences were designed to fold into one of two distinct topologies, A and B, with, respectively, low and high contact order [27]. The contact order of a given conformation is defined as the average sequence separation between amino acids that are in contact, and provides a measure of the complexity of the structure. It has been shown to be inversely correlated with the rate at which folding occurs for small single-domain real proteins [31]. In line with this, in kinetic explicit-chain Monte Carlo simulations, Faísca and Plaxco found that the sequences with topology A (1–3) indeed folded faster than those with topology B (4–6) [27].

B. QUBO representation

We wish to determine maximally compact lattice protein structures with minimum energy without having to resort to exhaustive enumeration of an exponentially large number of states. Unfortunately, conventional Monte Carlo methods for polymers are impractical for exploring chain structures on a minimal grid, since steric clashes render the acceptance probability very low, if not zero.

A possible way forward is to take a QUBO approach, where the chain is mapped onto a binary system and chain structure is enforced through penalty energies. For this, one may use a turn-based mapping, in which the bits encode the directions of the links along the chain [7,8]. However, it then becomes very cumbersome to implement nonlocal interactions along the chain, such as chain self-avoidance.

Instead, we use a fieldlike representation with bits at all lattice sites [9], which greatly facilitates the implementation of nonlocal interactions and thereby enables the study of longer chains. This representation uses more bits than what is needed to describe the very shape of a chain, which may seem wasteful. However, the total bit count is not worse than with turn-based methods [32], since no additional auxiliary bits are needed in order to have a quadratic energy function, as required on D-Wave's systems. Another fieldlike binary encoding of lattice proteins, though without the possibility of freezing the sequence, was recently developed by mapping the system to a lattice gauge theory [33].

Our fieldlike representation uses bits $b_{i,n}$ that indicate whether amino acid i is located on lattice site n ($b_{i,n} = 1$) or not ($b_{i,n} = 0$). The energy function is given by [9]

$$E = E_{\text{MJ}} + \lambda_1 E_1 + \lambda_2 E_2 \ + \lambda_3 E_3, \tag{2}$$

where E_{MJ} is the interaction potential [Eq. (1)] and the remaining three terms are penalty energies whose strengths are controlled by the Lagrange parameters λ_i. All terms have closed-form expressions valid for an arbitrary lattice and chain length N, which, in brief, are as follows [9].

(1) In a valid chain conformation, the interaction potential E_{MJ} [Eq. (1)] can be written as

$$E_{\text{MJ}} = -\sum_{|i-j|>1} C(a_i, a_j) \sum_{\langle n,m\rangle} b_{i,n} b_{j,m}, \tag{3}$$

where the second sum runs over all nearest-neighbor pairs n, m.

(2) The penalty energy E_1 is given by

$$E_1 = \sum_i \left(\sum_n b_{i,n} - 1 \right)^2, \tag{4}$$

and serves to ensure that each amino acid is located at exactly one lattice site.

(3) The penalty energy E_2 enforces chain self-avoidance and can be written as

$$E_2 = \frac{1}{2} \sum_n \sum_{i \neq j} b_{i,n} b_{j,n}. \tag{5}$$

(4) The penalty energy E_3 is responsible for chain connectivity and is given by

$$E_3 = \sum_{1 \leqslant i < N} \sum_n b_{i,n} \sum_{||m-n||>1} b_{i+1,m}, \tag{6}$$

where $||m - n||$ denotes the Eucledian distance between m and n (unit lattice spacing).

We minimize the QUBO energy E [Eq. (2)], using HA or SA, in order to minimize E_{MJ} [Eq. (1)] over chain conformations. For this to work, the Lagrange parameters λ_i must be above some thresholds. Beyond these thresholds, it turns out that the success rates of QUBO HA and QUBO SA are robust to small changes in the λ_i parameters [9] (see also Appendix A). In particular, this robustness enables us to use the same parameters, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3) = (1.5, 2.0, 2.0)$, for all six sequences studied (Table I).

C. HA

D-Wave offers access to solvers based solely on quantum annealing as well as hybrid quantum-classical solvers. The latter approach uses classical solvers while sending suitable subproblems as queries to the quantum processing unit (QPU). The solutions to the subproblems serve to guide the classical solvers [23]. The goal is to speed up the solution of challenging QUBO problems by queries to the QPU. The hybrid approach makes it possible to tackle problems with many thousands of fully connected variables, which is far beyond what currently can be dealt with using only the QPU.

Unless specified by the user, the D-Wave hybrid annealer selects the run-time based on the problem size. This default value is also the shortest run-time that can be used. This run-time may or may not be sufficiently long for satisfactory results [9]. For each of the six sequences in Table I, we did HA computations for one or more run-times. For each run-time, we generated a set of 100 runs. Starting with the default value (6 s for all six systems), we increased the run-time in steps of 2 s until all 100 runs returned the correct solution, which gave maximum required run-times varying between 6 and 22 s.

D. SA

A standard method for energy minimization in complex systems is Monte Carlo–based SA [22]. Here, one performs constant-temperature updates of the system at hand, as defined by the Boltzmann distribution $\propto e^{-\beta E}$, at a sequence of decreasing temperatures (β is inverse temperature).

We conducted SA computations for the six sequences in Table I, starting from random initial bit configurations. All runs spanned the same set of 25 geometrically distributed temperatures, given by $\beta_i = 1.05^i$ ($i = 1, \ldots, 25$). The updates were single-bit flips, controlled by a Metropolis acceptance criterion.

To assess the run-time dependence of the results, we generated runs comprising between 1000 and 80 000 sweeps per temperature, where one sweep corresponds to, on average, one attempted update per bit variable. One sweep required approximately 1.1 ms on an Intel Core i9-13900K processor. For each choice of sequence and run length, we performed a set of 100 runs, using different random number seeds.

E. GO

Besides HA and SA, we also tested using the GO solver [24], which combines simplex [34], branch-and-bound [35], and cutting-plane [36] methods with heuristics to optimize linear or quadratic objective functions under linear or quadratic constraints. Here, the penalty energies in Eq. (2) were replaced by hard constraints, whereas the binary representation of the systems remained the same as in our QUBO computations. In recent work on protein sequence optimization, the GO solver performed better than QUBO methods [37].

We ran the GO program locally under an academic license. For each sequence (Table I), we varied the run-time from 400 to 1200 s in steps of 200 s. For each combination of sequence and run-time, we performed ten runs using five threads each on an Intel Core i9-13900K. Adding more threads did not further improve the performance.

F. Enumeration of all structures

For our 48–amino acid sequences on a 4×4×3 grid, the number of distinct structures unrelated by symmetry is 134,131,827,475 [29,30].

To be able to compare our HA, SA, and GO results with exact data, we wrote and ran a straightforward C++ program implementing a depth-first graph search algorithm to generate these structures (see Appendix B). Using this data, we first confirmed the previously reported [27] minimum-energy structures. For each sequence, we then determined the density of states, calculated as a function of E_{MJ}. In addition, to compare the energy landscapes, we computed for each sequence the structural similarity with the minimum-energy state for the 100 lowest-energy states.

The calculations were done on the JUSUF cluster at the Jülich Supercomputing Centre, and required a total of 17 h of run-time on 512 processor cores (AMD EPYC 7742) for all six sequences.

III. RESULTS

Conventional sampling methods struggle with dense biomolecular systems, where chain motions are hampered by steric clashes. Here, we consider a minimal yet challenging example of such a system, namely, a single lattice protein confined on a minimal grid. Exploring the state space of this system with standard explicit-chain Monte Carlo methods is impractical, due to steric clashes. To overcome this issue, we adopt a binary representation (Sec. II B), which yields

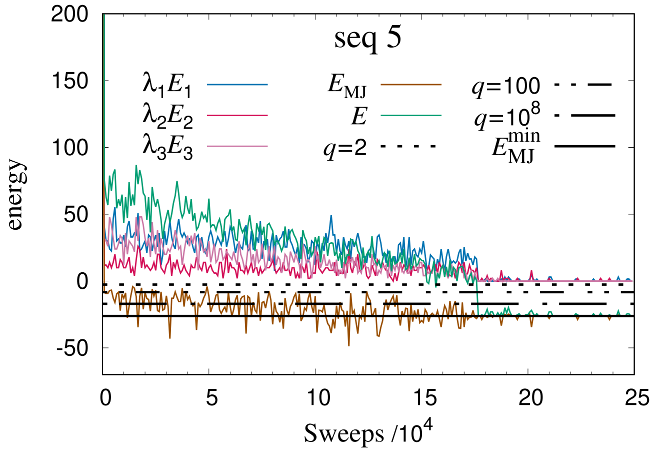

FIG. 1. Run-time evolution of the interaction potential $E_{\rm MJ}$ and the penalty energies E_1, E_2, and E_3 [Eqs. (3)–(6)] in an SA run for sequence 5 (Table I), using $\boldsymbol{\lambda} = (1.5, 2.0, 2.0)$, 25 temperatures, and 10 000 sweeps at each temperature. The horizontal lines indicate data on the distribution of $E_{\rm MJ}$ over valid chain structures, as obtained by exhaustive enumerations (Sec. II F). The lowest line represents the minimum $E_{\rm MJ}$, $E_{\rm MJ}^{\rm min}$. The other three lines represent the lowest q quantiles for $q = 2$ (median), $q = 100$, and $q = 10^8$, respectively.

an expanded state space and also opens it up for quantum optimization. Specifically, we use this representation to minimize the energy of the lattice protein, as given by a pairwise Miyazawa-Jernigan potential ($E_{\rm MJ}$; Sec. II A). We test three methods for solving this binary optimization task. The first two, HA (Sec. II C) and SA (Sec. II D), are QUBO methods, where chain structure is enforced through penalty energies (E_1,E_2,E_3). The third method, GO (Sec. II E), works with hard constraints. As a test bed, we use a set of six 48–amino acid sequences (Table I), all of which we study on a 4×4×3 grid.

Figure 1 shows the run-time evolution of the interaction potential $E_{\rm MJ}$ and the penalty energies E_1, E_2, and E_3 in an SA run for sequence 5 (Table I), using 25 temperatures and 10^4 Monte Carlo sweeps (Sec. II D) at each temperature.

The acceptance rate for the Metropolis update is modest, varying from ≈0.01 at the higher temperatures to $\approx 5\times10^{-5}$ at the lowest temperature. Nevertheless, the system is capable of evolving toward more chainlike structures with lower penalty energies, and toward the known minimum $E_{\rm MJ}$ value. The correct minimum-energy conformation is indeed found.

For reference, Fig. 1 shows the lowest q quantiles of the $E_{\rm MJ}$ probability distribution for $q = 2$ (median), $q = 100$, and $q = 10^8$, as obtained by exhaustive enumeration of all structures (Sec. II F). The system visits the correct solution for the first time while at the 18th temperature. After this point, the system is occasionally found in unphysical states, but it stays close to the correct solution and well below the lowest 10^8 quantile in terms of $E_{\rm MJ}$.

A. Comparing HA, SA, and GO

We wish to assess the ability of the HA, SA, and GO methods to locate minimum-$E_{\rm MJ}$ chain conformations, and its run-time dependence, for the six sequences in Table I. For a given sequence and run-time, we generate a set of 10 (GO) or 100 (HA and SA) independent runs. We denote the final energy E [Eq. (2)] in a given run by E_f, and the average of E_f over the independent runs by $\overline{E_f}$. The success rate is defined as the fraction of the runs that return the correct solution, $E_{\rm MJ}^{\rm min}$. Determining a 100% success rate is equivalent to determining that $\overline{E_f} = E_{\rm MJ}^{\rm min}$.

Figure 2 shows the run-time dependence of the average final energy in our HA, SA, and GO computations. At the longest run-times, $\overline{E_f}$ is close or equal to $E_{\rm MJ}^{\rm min}$ and falls well below the lowest 10^8 quantile of the $E_{\rm MJ}$ distribution for all six sequences with both HA and SA. In fact, with HA, the success rate is 1 for all six sequences. With SA, the success rate is 1 for the sequences 4–6, while falling between 0.87 and 0.98 for the sequences 1–3. Thus, the sought solutions can be readily found with both HA and SA. By contrast, our GO computations failed to find any correct solutions. Indeed, comparing the classical GO and SA methods for a given run-time, $\overline{E_f}$ is much higher with GO than SA (Fig. 2).

It is worth noting that the HA success rate is $\geqslant$0.6 for all six sequences even when using the shortest run-time possible on D-Wave's system (6 s). In previous work on 2D lattice proteins [9], the run-time had to be increased above a system-dependent threshold in order to attain a satisfactory success rate. For the systems studied in the present paper, it seems that the shortest possible run-time is above any such threshold.

For a given sequence, starting with the default run-time (6 s), we increased the HA run-time in steps of 2 s until all 100 runs returned the correct solution. With this procedure, we found that the longest required run-time varied between 6 and 22 s, depending on sequence.

In SA, we used a fixed set of run-times for all sequences. The longest runs, corresponding to 80 000 Monte Carlo sweeps per temperature, took 2.1×10^3 s using one thread on an Intel Core i9-13900K processor. This run-time gave a success rate of one (sequences 4–6), or slightly below one (sequences 1–3; success rates between 0.87 and 0.98).

In sharp contrast to the HA and SA results, we were unable to find low-energy structures with the GO method (Fig. 2). In fact, the success rate was zero for all sequences and run-times used. This poor performance can be traced to the chain connectivity constraint, encoded by the penalty energy E_3 in Eq. (2), which, unlike the other two constraints, cannot be expressed in terms of linear equalities or inequalities. If this constraint is removed, one has a multicomponent lattice gas of amino acids, rather than a protein chain. For this gas, we found that GO worked very well. Having seen this, we also tried including the E_3 term in the GO objective function, instead of treating chain connectivity as a hard constraint. However, the results were similarly poor. In either case, virtually all improvements of the objective function in the GO runs occurred in steps based on heuristics. To be able to successfully use GO for this problem, it appears that a different binary representation would have to be devised.

B. Comparing HA and SA data with exact results

The number of maximally compact chain conformations on a 4×4×3 lattice (chain length $N = 48$), $\approx 1.3\times10^{11}$, is known from previous work [29,30]. Using exhaustive enumeration techniques (Sec. II F and Appendix B), we extended this work to obtain the energies of all structures and thereby

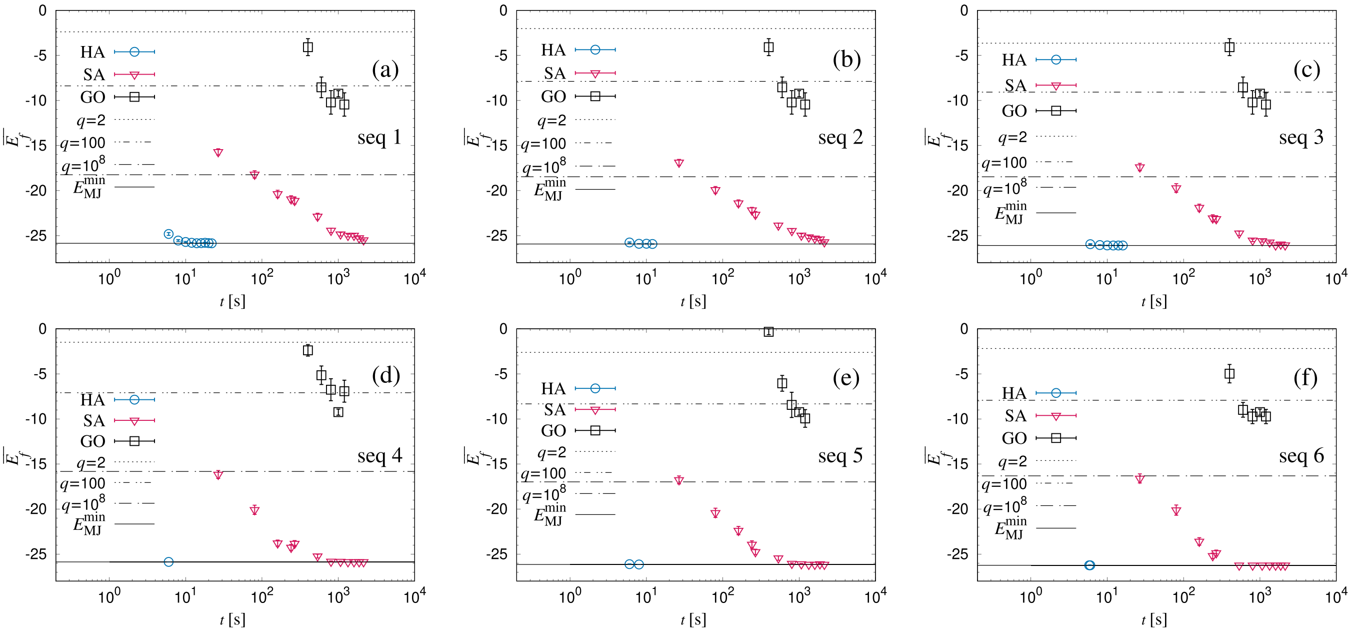

FIG. 2. Average final energy, $\overline{E_f}$, plotted against run-time, t, in HA, SA, and GO computations for the six sequences in Table I, with Lagrange parameters $\boldsymbol{\lambda} = (1.5, 2.0, 2.0)$ [Eq. (2)]. Each data point represents an average over 10 (GO) or 100 (HA and SA) independent runs. The horizontal lines indicate data on the distribution of E_{MJ} over valid chain structures, as obtained by exhaustive enumerations (Sec. II F). The lowest line represents $E_{\mathrm{MJ}}^{\mathrm{min}}$. The other three represent the lowest q quantiles for $q = 2$ (median), $q = 100$, and $q = 10^8$, respectively. (a) Sequence 1. (b) Sequence 2. (c) Sequence 3. (d) Sequence 4. (e) Sequence 5. (f) Sequence 6.

the exact density of states, $g(E_{\mathrm{MJ}})$, for the sequences in Table I. These results provide us with a ground truth with which the HA and SA results can be compared. Generating similar results for a larger chain length would be challenging due to the large processor and storage requirements. Therefore, the chain length $N = 48$ is well suited for testing the power of the QUBO approach.

Figure 3 shows histograms of the final energy E_f in the HA and SA runs for our shortest and longest run-times for the different sequences. Also shown, for comparison, are the densities of states, $g(E_{\mathrm{MJ}})$ (insets). The $g(E_{\mathrm{MJ}})$ data has an effectively continuous and approximately Gaussian central part where the bulk of the states are found, with an inflection point on the low-energy arm of the curve (at $E_{\mathrm{MJ}} \approx -17$). The

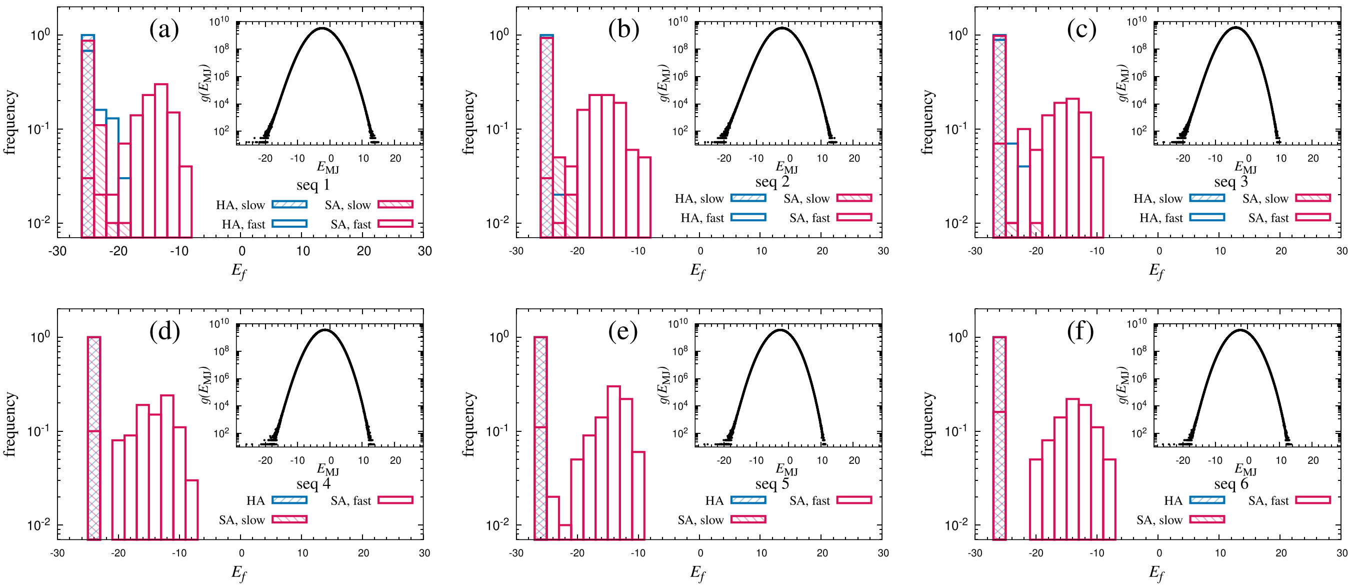

FIG. 3. Histograms of the final energy E_f [Eq. (2)] on a logarithmic scale for HA and SA computations for the six sequences in Table I, for the shortest (fast) and longest (slow) run-times used. The insets show the density of states (logarithmic scale) calculated as a function of E_{MJ}. (a) Sequence 1. (b) Sequence 2. (c) Sequence 3. (d) Sequence 4. (e) Sequence 5. (f) Sequence 6.

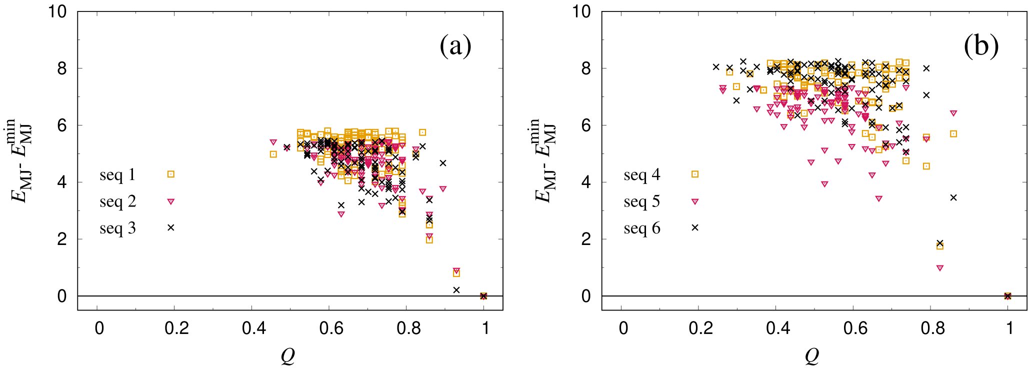

FIG. 4. Energy landscapes showing $E_{\text{MJ}} - E_{\text{MJ}}^{\text{min}}$ against the overlap Q with the native state for the 100 lowest-lying maximally compact states for each of the six sequences in Table I, as obtained through exhaustive enumeration (Sec. II F and Appendix B). The overlap, or nativeness, Q of a structure is the fraction of ground state contacts it contains. (a) Sequences 1–3, with low-complexity topology A as their native state. (a) Sequences 4–6, with high-complexity topology B as their native state.

states of main interest to us are the lowest-lying ones, which form a discrete tail of the distribution.

The final energies E_f of the HA runs are consistently located in the discrete low-energy tail of $g(E_{\text{MJ}})$, even for the shortest run-time (6 s). Increasing this annealing time to around four times the minimum allowed value results in $\overline{E_f} = E_{\text{MJ}}^{\text{min}}$ for the HA runs. For SA with the shortest run-time (27 s), a significant fraction of the runs have final energies E_f in the continuous part of the distribution. However, increasing the run-time shifts the E_f distribution downward, as it should. At the longest run-time (2.1×10^3 s), most of the runs end up in the desired minimum-E_{MJ} state.

The ground state is given by topology A for three of the sequences (1–3) and by topology B for the other three sequences (4–6). It is worth noting that the QUBO-based HA and SA methods solve the energy minimization problem faster for the sequences associated with topology B than for those associated with topology A (Fig. 2). By contrast, in explicit-chain Monte Carlo simulations of the same sequences on an unrestricted grid, Faísca and Plaxco found that folding occurred faster for the low-complexity topology A than for the high-complexity topology B [27]. To shed light on what causes this difference between QUBO and explicit-chain results, we generated energy landscapes showing energy against nativeness for the 100 lowest-lying maximally compact states (Fig. 4). Here, the nativeness of a given structure is defined as the fraction of ground state contacts present in the structure.

Figure 4 shows the computed energy landscapes. While it is important to remember that this analysis is restricted to maximally compact states, it is interesting that the sequences associated with the respective topologies A and B have notably different energy landscapes. For the former [Fig. 4(a)], the data hints at a dense and funnellike organization of the low-lying states. This property may facilitate finding the native state with conventional explicit-chain methods [38] featuring diffusive state space exploration, which is consistent with the relatively fast folding of these sequences observed by Faísca and Plaxco on an unrestricted grid [27]. The high barriers between the sparsely organized low-lying states of the high complexity systems [Fig. 4(b)] pose a substantially harder challenge for such approaches. With QUBO HA or SA the exploration of state space happens over an extended conformation space containing additional unphysical states that seem to facilitate the search for low-lying valid states for all our sequences. The clear separation in energy between the native and competing states seems to facilitate the discovery of the ground states for sequences 4–6 in the QUBO-based methods (Figs. 2 and 4).

IV. DISCUSSION

Computational modeling of dense biomolecular systems is important to better understand the properties of individual proteins under cellular conditions, and biophysical processes such as protein aggregation and phase separation. A major obstacle in studying such problems with conventional explicit-chain methods is steric clashes, which tend to make the exploration of state space very slow. As a simple yet extreme example of a dense system, here we have considered the task of determining minimum-energy structures of lattice proteins confined on a minimal grid, without any free sites.

The results presented show that this problem can be swiftly and consistently solved by recasting it as a QUBO problem and using HA or SA. In fact, with HA, it was possible to find the correct solution for chains with length $N = 48$ in the order of 10 s (Sec. III A).

Traditionally, the problem of finding maximally compact lattice protein states with minimum energy has been tackled through brute-force exhaustive enumerations. We implemented and used this approach for our 4×4×3 system. However, the calculations required a significant amount of time (approximately 17 h on 512 AMD EPYC 7742 processor cores) and storage, and would be cumbersome to extend to longer chains due to the exponential growth of state space.

Another alternative is to stick to the binary description of the QUBO formulation, but replace the soft constraints in favor of hard ones and use optimization methods based on linear, quadratic, or constraint programming. This approach was recently tested on two optimization problems somewhat related to ours, namely, lattice protein design [37] and peptide docking [39]. In both cases, these methods showed good results. Therefore, we tested using the GO method for our problem. In stark contrast to the good results obtained with this method for the design problem [37], GO failed to find satisfactory solutions to our problem (Sec. III A). To understand this poor performance, we performed additional GO computations for systems where the chain connectivity constraint, as encoded by the penalty energy E_3, was left out. For this system, corresponding to a heterogeneous lattice gas, we found that GO worked very well. Hence, it appears that the poor results obtained with GO for our lattice protein problem can be attributed to the presence of the chain connectivity constraint. We note that this constraint, unlike the two encoded by the penalty energies E_1 and E_2, cannot be expressed in terms of linear equalities or inequalities, and it is absent in the design problem [37].

The folding rate of single-domain proteins is known to be inversely correlated with the contact order of the native structure [31]. In line with this, previous work found folding to be faster for our sequences 1–3 than for 4–6 (Table I) in kinetic explicit-chain Monte Carlo simulations on an unrestricted grid [27]. In our QUBO-based HA and SA computations, the opposite trend is observed; the ground state is easier to locate for the sequences 4–6 than for 1–3 (Fig. 2). This finding underscores that the extended QUBO systems need not follow intuition based on explicit-chain dynamics.

V. CONCLUSION AND OUTLOOK

We have tested using a QUBO formulation, amenable to quantum optimization, to solve the simplified yet computationally challenging problem of determining maximally compact minimum-energy structures of lattice proteins. The results presented show that this problem can be swiftly and successfully solved with the QUBO approach using either classical (SA) or hybrid quantum-classical (HA) computations, for a chain length of 48. In fact, with HA, it took the order of 10 s to obtain correct solutions. We are not aware of any existing competing method with a potentially similar performance.

The QUBO-based computations presented here can be easily extended to dense multichain systems, as in protein aggregation [40] or phase separation [41]. In their present form, the methods assume, however, a lattice-based biophysical model. Whether or not they can be extended to continuous models remains to be seen.

The QUBO approach, which was employed for optimization problems already in the 1980s [10,11], may be relevant for problems seemingly different from the ones studied here. Since it opens up a nonphysical extension to the state space, it can be utilized to "walk around" high-energy barriers. In particular, other problems where constraint-fulfilling updates are difficult, such as tight scheduling problems [19–21], might benefit from such an encoding.

ACKNOWLEDGMENTS

We thank Carsten Peterson for fruitful discussions. We gratefully acknowledge the Jülich Supercomputing Centre for supporting this project by providing computing time on the D-Wave Advantage$^{\text{TM}}$ System JUPSI through the Jülich UNified Infrastructure for Quantum computing (JUNIQ).

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

APPENDIX A: CHOICE OF LAGRANGE PARAMETERS

The QUBO energy, Eq. (2), contains three Lagrange parameters, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$, which need to be set. We investigated the dependence of the success rate on these parameters in both QUBO HA and QUBO SA through computations for sequence 5 (Table I). Here, we varied one λ_i at a time in steps of 0.25, while keeping the other two fixed. The highest success rates were obtained using $\boldsymbol{\lambda}^* = (1.5, 2.0, 2.0)$.

Figure 5 illustrates the sensitivity of the success rate to changes in individual λ_i parameters. Fortunately, the success rate is robust to small changes in these parameters. Because of this robustness and the fact that our sequences share a common length, we decided to use the same parameters for all six sequences, which worked well.

APPENDIX B: ENUMERATION OF ALL STRUCTURES

We wish to enumerate all the $\approx 1.3\times 10^{11}$ [29,30] directed Hamiltonian paths unrelated by symmetry on a $4\times 4\times 3$ lattice. There is a total of 16 symmetry operations on this lattice, including the identity [30]. Let (x, y, z) denote lattice coordinates, where $x, y \in \{0, 1, 2, 3\}$ and $z \in \{0, 1, 2\}$.

By symmetry, there are six possible starting points for the paths, which may be taken as $(0, 0, 0)$, $(1, 0, 0)$, $(1, 1, 0)$, $(0, 0, 1)$, $(1, 0, 1)$, and $(1, 1, 1)$. After selecting a starting point, one or both of two symmetries may remain unbroken. One remaining symmetry operation is reflection across the (x, x, z) plane, which leaves the starting points $(0, 0, 0)$, $(0, 0, 1)$, $(1, 1, 0)$, and $(1, 1, 1)$ unchanged. To break this symmetry, we note that when a path starting at (x_0, y_0, z_0) leaves the line (x_0, y_0, z) for the first time, it does so in one of four directions: $\pm\hat{\mathbf{x}}$ and $\pm\hat{\mathbf{y}}$, where $+\hat{\mathbf{x}}, +\hat{\mathbf{y}}$ and $-\hat{\mathbf{x}}, -\hat{\mathbf{y}}$ are symmetry-related pairs. The symmetry is broken by allowing only one direction from each pair, say $+\hat{\mathbf{x}}$ and $-\hat{\mathbf{y}}$. The other remaining symmetry operation is reflection across the $(x, y, 1)$ plane, which leaves the starting points $(0, 0, 1)$, $(0, 0, 1)$, and $(1, 1, 1)$ unchanged. This symmetry is broken by stipulating that the first step away from the $(x, y, 1)$ plane is in, say, the $-\hat{\mathbf{z}}$ direction.

To generate the desired paths, we proceed in two steps. The first step is a width-first search, where symmetries are accounted for by using the starting points and rules just described. Given all self-avoiding paths with length $n - 1$, we generate and store all with length n. This is iterated up to path length $n = 20$. This is a very memory-intensive approach. In the second step, we therefore switch to a depth-first search to add the remaining 28 nodes to the paths. For each of the seed paths obtained in the previous step, we recursively extend the structure until we exhaust all lattice sites or until we reach a dead end. The heart of the procedure is a function to explore all extensions of a partial path of length

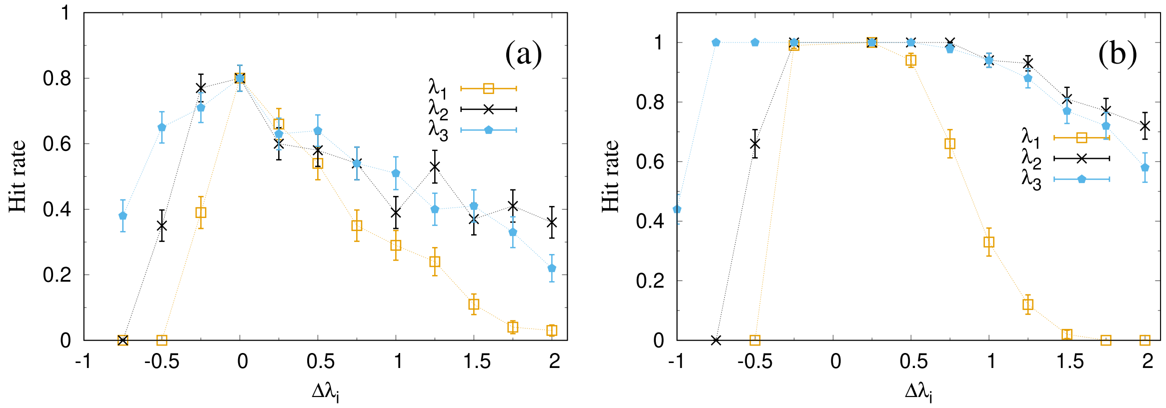

FIG. 5. Parameter dependence of the fraction of correct solutions (hit rate) in the vicinity of the best Lagrange parameters found, $\boldsymbol{\lambda}^* = (1.5, 2.0, 2.0)$, when using QUBO SA and QUBO HA to search for the ground state of sequence 5 (Table I in the main text). The hit rate is plotted against $\Delta\lambda_i = \lambda_i - \lambda_i^*$, keeping $\lambda_j = \lambda_j^*$ for $j \neq i$. Each data point represents an average over 100 runs. The SA runs comprised 10 000 sweeps per temperature, while the HA run-time was 10 s. Lines are drawn to guide the eye. (a) SA. (b) HA.

n. Given an input path of length n, the function loops over all neighbors of the last node in the input path. If a neighbor appears anywhere in the path up to position $n-1$, it is discarded. Otherwise, we create a new path by adding it to the input path. We then recursively call the same function to explore all extensions of the incremented path. Whenever a path length $n = 48$ is given as an input to the function, it stores it, and returns without any further recursive calls. The simple loop and recursion algorithm above presents easy opportunities for parallelization. We used OneAPI Threading Building Blocks (onetbb) for parallelization of the structure search.

Storage and postprocessing of the very large number of structures in problems like this present their own challenges. We devised an obvious compact representation for the explored paths, requiring 48 bytes per path. Despite this, the files with explored paths grew to many terabytes in size. However, our recursive procedure stores similar paths close to each other in the file, which allows for very large compression factors to be used (for instance with `xz`) for long-term storage.

[1] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back *et al.*, Highly accurate protein structure prediction with AlphaFold, Nature (London) **596**, 583 (2021).

[2] K. F. Lau and K. A. Dill, A lattice statistical mechanics model of the conformational and sequence spaces of proteins, Macromolecules **22**, 3986 (1989).

[3] A. Šali, E. Shakhnovich, and M. Karplus, Kinetics of protein folding: A lattice model study of the requirements for folding to the native state, J. Mol. Biol. **235**, 1614 (1994).

[4] N. D. Socci and J. N. Onuchic, Folding kinetics of proteinlike heteropolymers, J. Chem. Phys. **101**, 1519 (1994).

[5] D. K. Klimov and D. Thirumalai, Criterion that determines the foldability of proteins, Phys. Rev. Lett. **76**, 4070 (1996).

[6] H. Li, R. Helling, C. Tang, and N. Wingreen, Emergence of preferred structures in a simple model of protein folding, Science **273**, 666 (1996).

[7] A. Perdomo-Ortiz, N. Dickson, M. Drew-Brook, G. Rose, and A. Aspuru-Guzik, Finding low-energy conformations of lattice protein models by quantum annealing, Sci. Rep. **2**, 571 (2012).

[8] A. Robert, P. K. Barkoutsos, S. Woerner, and I. Tavernelli, Resource-efficient quantum algorithm for protein folding, npj Quantum Inf. **7**, 38 (2021).

[9] A. Irbäck, L. Knuthson, S. Mohanty, and C. Peterson, Folding lattice proteins with quantum annealing, Phys. Rev. Res. **4**, 043013 (2022).

[10] J. J. Hopfield and D. W. Tank, Neural computation of decisions in optimization problems, Biol. Cybern. **52**, 141 (1985).

[11] C. Peterson and J. R. Anderson, Neural networks and NP-complete problems; a performance study of the graph bisectioning problem, Complex Syst. **2**, 59 (1988).

[12] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, Phys. Rev. E **58**, 5355 (1998).

[13] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem, Science **292**, 472 (2001).

[14] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[15] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, Algorithms **12**, 34 (2019).

[16] N. Mohseni, P. L. McMahon, and T. Byrnes, Ising machines as hardware solvers of combinatorial optimization problems, Nat. Rev. Phys. **4**, 363 (2022).

[17] C. Micheletti, P. Hauke, and P. Faccioli, Polymer physics by quantum computing, Phys. Rev. Lett. **127**, 080501 (2021).

[18] F. Slongo, P. Hauke, P. Faccioli, and C. Micheletti, Quantum-inspired encoding enhances stochastic sampling of soft matter systems, Sci. Adv. **9**, eadi0204 (2023).

[19] M. Lagerholm, C. Peterson, and B. Söderberg, Airline crew scheduling with Potts neurons, Neural Comput. **9**, 1589 (1997).

[20] D. Venturelli, D. J. J. Marchand, and G. Rojo, Quantum annealing implementation of job-shop scheduling, arXiv:1506.08479.

[21] T. Stollenwerk, S. Hadfield, and Z. Wang, Toward quantum gate-model heuristics for real-world planning problems, IEEE Trans. Quantum Eng. **1**, 1 (2020).

[22] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, Science **220**, 671 (1983).

[23] C. McGeoch, P. Farré, and W. Bernoudy, *D-Wave Hybrid Solver Service + Advantage: Technology Update*, Technical Report No. 14-1048A-A, D-Wave Systems Inc., 2020.

[24] Gurobi Optimization, LLC, Gurobi optimizer reference manual (2024), https://www.gurobi.com/.

[25] S. Miyazawa and R. L. Jernigan, Estimation of effective interresidue contact energies from protein crystal structures: Quasi-chemical approximation, Macromolecules **18**, 534 (1985).

[26] S. Miyazawa and R. L. Jernigan, Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading, J. Mol. Biol. **256**, 623 (1996).

[27] P. F. Faísca and K. W. Plaxco, Cooperativity and the origins of rapid, single-exponential kinetics in protein folding, Protein Sci. **15**, 1608 (2006).

[28] A. Irbäck, L. Knuthson, S. Mohanty, and C. Peterson, Using quantum annealing to design lattice proteins, Phys. Rev. Res. **6**, 013162 (2024).

[29] V. S. Pande, C. Joerg, A. Y. Grosberg, and T. Tanaka, Enumerations of the Hamiltonian walks on a cubic sublattice, J. Phys. A: Math. Gen. **27**, 6231 (1994).

[30] A. Kloczkowski and R. Jernigan, Computer generation and enumeration of compact self-avoiding walks within simple geometries on lattices, Comput. Theor. Polym. Sci. **7**, 163 (1997).

[31] K. W. Plaxco, K. T. Simons, and D. Baker, Contact order, transition state placement and the refolding rates of single domain proteins, J. Mol. Biol. **277**, 985 (1998).

[32] H. Linn, I. Lyngfelt, L. García-Álvarez, and G. Johansson, Resource analysis of quantum algorithms for coarse-grained protein folding models, Phys. Rev. Res. **6**, 033112 (2024).

[33] V. Panizza, A. Roggero, P. Hauke, and P. Faccioli, Statistical mechanics of heteropolymers from lattice gauge theory, Phys. Rev. Lett. **134**, 158101 (2025).

[34] G. B. Dantzig, *Linear Programming and Extensions* (Princeton University Press, Princeton, 1963).

[35] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning, Discrete Optim. **19**, 79 (2016).

[36] H. Marchand, A. Martin, R. Weismantel, and L. Wolsey, Cutting planes in integer and mixed integer programming, Discrete Appl. Math. **123**, 397 (2002).

[37] V. Panizza, P. Hauke, C. Micheletti, and P. Faccioli, Protein design by integrating machine learning and quantum-encoded optimization, PRX Life **2**, 043012 (2024).

[38] J. D. Bryngelson, J. N. Onuchic, N. D. Socci, and P. G. Wolynes, Funnels, pathways, and the energy landscape of protein folding: A synthesis, Proteins **21**, 167 (1995).

[39] J. K. Brubaker, K. E. C. Booth, A. Arakawa, F. Furrer, J. Ghosh, T. Sato, and H. G. Katzgraber, Quadratic unconstrained binary optimization and constraint programming approaches for lattice-based cyclic peptide docking, arXiv:2412.10260.

[40] S. Abeln, M. Vendruscolo, C. M. Dobson, and D. Frenkel, A simple lattice model that captures protein folding, aggregation and amyloid formation, PLoS ONE **9**, e85185 (2014).

[41] U. Rana, C. P. Brangwynne, and A. Z. Panagiotopoulos, Phase separation vs aggregation behavior for model disordered proteins, J. Chem. Phys. **155**, 125101 (2021).